УДК 004.415.2:004.8

Шелухин Я.И.

студент

Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ФРЕЙМВОРКОВ МОБИЛЬНОГО МАШИННОГО ОБУЧЕНИЯ НА ANDROID-УСТРОЙСТВАХ

Аннотация: В условиях активного развития мобильных технологий применение методов машинного обучения непосредственно на мобильных устройствах приобретает всё большую актуальность. В статье представлен сравнительный анализ современных фреймворков мобильного машинного обучения TensorFlow Lite, PyTorch Mobile и MNN. Рассмотрены их особенности, преимущества и ограничения, а также сформулированы рекомендации по выбору фреймворка в зависимости от требований мобильного приложения.

*Ключевые слова***:** машинное обучение, TensorFlow Lite, PyTorch Mobile, мобильные платформы, инференс на устройстве, мобильные приложения, искусственный интеллект, нейронные сети.

Shelukhin Y. I.

Student

Moscow Polytechnic University

COMPARATIVE ANALYSIS OF MOBILE MACHINE LEARNING FRAMEWORKS ON ANDROID DEVICES

Abstract: In the context of the rapid development of mobile technologies, the application of machine learning methods directly on mobile devices is becoming increasingly relevant. This paper presents a comparative analysis of modern mobile machine learning frameworks, including TensorFlow Lite, PyTorch Mobile, and MNN. Their features, advantages, and limitations are discussed, and recommendations for framework selection depending on mobile application requirements are formulated.

Keywords: machine learning, TensorFlow Lite, PyTorch Mobile, mobile platforms, on-device inference, mobile applications, artificial intelligence, neural networks.

Введение

Машинное обучение (machine learning, ML) является одним из наиболее быстро развивающихся направлений в области искусственного интеллекта[1], и его применение в мобильных приложениях открывает новые возможности для создания инновационных сервисов и функций. Мобильные устройства, такие как смартфоны и планшеты, становятся все более мощными и способными к выполнению сложных вычислений, что делает возможным использование машинного обучения в мобильных приложениях для решения таких задач, как применение компьютерного зрения, обработка естественного языка, предсказательная аналитика и многих других. Вычисления на устройстве пользователя стали особенно актуальны в последнее время, связи с отсутствием возможности их выполнения на стороне сервера из-за ограничений связи.

Однако, разработка ML-приложений для мобильных устройств сопряжена с рядом особенностей, таких как: значительные ограничения по ресурсам, энергопотреблению и необходимым уровнем безопасности.

Для преодоления этих сложностей были разработаны специализированные фреймворки, которые позволяют эффективно использовать ML в мобильных приложениях.

В настоящее время существует несколько популярных фреймворков для мобильного ML, каждый из которых имеет свои сильные и слабые стороны. В этой статье будет проведено сравнение четырех наиболее популярных фреймворков: TensorFlow Lite, Core ML, PyTorch Mobile и MNN (Mobile Neural Network).

Целью данной работы является сравнительный анализ производительности популярных фреймворков мобильного машинного обучения при выполнении задачи классификации изображений на Android-устройстве.

Для достижения поставленной цели были решены следующие задачи:

1. Выполнен анализ особенностей современных мобильных ML-фреймворков.
2. Разработана методика тестирования мобильных ML-фреймворков.
3. Реализованы тестовые мобильные приложения и проведено экспериментальное сравнение следующих фреймворков: TensorFlow Lite, PyTorch Mobile и MNN.
4. Сформулированы рекомендации по выбору фреймворка для мобильных ML-приложений.

Анализ фреймворков мобильного машинного обучения

На мобильных устройствах алгоритмы машинного обучения интегрируются в приложения с использованием специализированных

фреймворков, обеспечивающих выполнение нейросетевых моделей в условиях ограниченных вычислительных ресурсов. Такие фреймворки предоставляют средства оптимизации моделей, аппаратного ускорения вычислений и взаимодействия с мобильными платформами. В настоящее время одними из наиболее распространённых решений являются TensorFlow Lite, PyTorch Mobile и MNN, получившие широкое распространение благодаря поддержке Android и iOS, высокой производительности и активному развитию со стороны сообщества и крупных технологических компаний [2]. Далее будут рассмотрены особенности указанных фреймворков и выполнен их сравнительный анализ.

TensorFlow Lite представляет собой облегченную версию популярного фреймворка TensorFlow, специально разработанная для использования на мобильных устройствах и встраиваемых устройствах. Основными преимуществами TensorFlow Lite являются широкая экосистема, поддержка оптимизации моделей и наличие большого количества инструментов для конвертации и развертывания нейронных сетей [3].

Фреймворк поддерживает Android и iOS, а также предоставляет механизмы аппаратного ускорения вычислений посредством GPU и специализированных нейронных процессоров.

К преимуществам TensorFlow Lite можно отнести:

- развитую документацию;
- широкое сообщество разработчиков;
- поддержку квантизации моделей;
- совместимость с экосистемой TensorFlow.

Основным недостатком является сравнительно большой размер встраиваемых библиотек и необходимость дополнительной оптимизации моделей для достижения высокой производительности.

PyTorch Mobile является мобильной версией популярного фреймворка PyTorch для машинного обучения. PyTorch Mobile позволяет выполнять модели, обученные с помощью PyTorch, на мобильных устройствах. Основным преимуществом данного решения является простота интеграции моделей, разработанных в экосистеме PyTorch [3].

Фреймворк поддерживает платформы Android и iOS, а также предоставляет возможность выполнения нейросетевых моделей непосредственно на устройстве пользователя.

Преимуществами PyTorch Mobile являются:

- совместимость с PyTorch;
- активная поддержка сообщества.

К недостаткам можно отнести сравнительно большой размер runtime-библиотек и более низкую производительность по сравнению со специализированными мобильными решениями.

MNN – это фреймворк для мобильного машинного обучения, разработанный компанией Alibaba. Основной особенностью MNN является, то, что он ориентирован на высокую производительность и энергоэффективность на мобильных устройствах. Данный фреймворк поддерживает как Android, так и iOS.

К преимуществам MNN относятся:

- высокая скорость инференса;
- компактный размер библиотеки;
- эффективное использование аппаратных ресурсов.

В свою очередь у него есть и свои недостатки: слабая документация и малое количество проектов, которые возможно использовать как образцы кода, меньшее сообщество разработчиков по сравнению с другими фреймворками.

Каждый из рассмотренных в работе фреймворков имеет свои особенности и предназначен для решения различных задач в области мобильного машинного обучения. Выбор наиболее подходящего фреймворка зависит от конкретных требований проекта, к которым могут относится следующие факторы: поддерживаемые платформы, производительность, размер модели и другие.

Методика эксперимента

Для проведения сравнительного анализа была реализована серия тестовых мобильных приложений для Android, использующих TensorFlow Lite, PyTorch Mobile и MNN.

Во всех тестах использовался только CPU-инференс без применения GPU-ускорения на следующей аппаратной платформе, представляющей собой смартфон Umidigi Bison, который имеет нижеприведенные с характеристики:

- операционная система Android 10;
- процессор MediaTek Helio P60;
- 6 ГБ оперативной памяти.

В качестве тестовой модели машинного обучения для всех фреймворков использовалась одинаковая модель классификации изображений: MobileNetV2 – компактная сверточная нейронная сеть, широко применяемая в мобильных ML-приложениях благодаря высокому соотношению точности и вычислительной эффективности [4].

Тестовая выборка была составлена с использованием 700 случайно выбранных изображений из набора данных ImageNet. Для каждого из изображений выполнялся инференс модели, после чего была взята медиана времени обработки для каждого из фреймворков.

В рамках эксперимента анализировались следующие параметры:

- среднее время инференса;
- итоговый размер мобильного приложения.

Результаты эксперимента

Результаты проведения сравнительного тестирования представлены в таблице 1.

Таблица 1. Сравнение фреймворков для мобильного машинного обучения

Фреймворк	**Среднее время инференса**	**Размер APK**	**Top-1 точность**
TensorFlow Lite	73 мс	27 МБ	71.86 %
PyTorch Mobile	137 мс	163 МБ	71.43 %
MNN	69 мс	35 МБ	71.71 %

Полученные данные показывают, что наименьшее время инференса продемонстрировал фреймворк MNN. Это связано с высокой степенью оптимизации вычислительных операций для мобильных процессоров.

TensorFlow Lite показал близкие результаты по скорости выполнения, при этом обеспечивая меньший размер приложения по сравнению с PyTorch Mobile.

PyTorch Mobile продемонстрировал наиболее низкую производительность и максимальный размер итогового приложения.

Вероятной причиной является больший объем runtime-компонентов и меньшая степень оптимизации для мобильных устройств.

Разница в точности между фреймворками незначительна и вероятно будет сглажена на большой выборке данных.

Анализ результатов

Результаты эксперимента демонстрируют, что выбор фреймворка мобильного машинного обучения существенно влияет на производительность и размер итогового приложения.

MNN обеспечивает минимальное время инференса, что делает данный фреймворк перспективным для задач, требующих высокой скорости обработки данных на мобильном устройстве.

TensorFlow Lite демонстрирует наиболее сбалансированные характеристики. Несмотря на несколько большее время инференса по сравнению с MNN, данный фреймворк обладает развитой экосистемой, поддержкой большого количества инструментов и высокой степенью зрелости.

PyTorch Mobile в большей степени ориентирован на исследовательские задачи и удобство переноса моделей из экосистемы PyTorch, однако в условиях мобильных устройств уступает специализированным решениям по производительности и размеру runtime.

В ходе обзора популярных фреймворков для мобильного машинного обучения - TensorFlow Lite, Core ML, PyTorch Mobile и MNN - были рассмотрены их особенности, преимущества и недостатки. Каждый из фреймворков имеет свои уникальные характеристики, которые делают их подходящими для различных типов проектов и задач.

Экспериментальные результаты показали, что:

- MNN обеспечивает минимальное время инференса;
- TensorFlow Lite демонстрирует оптимальный баланс между производительностью и удобством разработки;
- PyTorch Mobile характеризуется большим размером итогового приложения и более низкой производительностью.

Следует отметить, что результаты исследования ограничены использованием одного устройства и только CPU-инференса. Производительность фреймворков может изменяться при использовании GPU или NPU-ускорения, а также на других аппаратных платформах.

Используя результаты проведенных экспериментов и выполненный анализ, можно сформулировать следующие рекомендации по выбору мобильного ML-фреймворка.

TensorFlow Lite выделяется широкой экосистемой и хорошей документацией, что делает его привлекательным выбором для разработчиков.

PyTorch Mobile, с удобством использования для разработчиков, знакомых с PyTorch, и небольшим размером библиотеки, представляет интерес для тех, кто предпочитает этот фреймворк.

MNN, ориентированный на высокую производительность и энергоэффективность, может быть хорошим выбором для проектов, где эти параметры критичны.

Полученные результаты могут быть использованы при выборе фреймворка для разработки мобильных приложений с поддержкой машинного обучения.

Заключение

Таким образом, были определены особенности применения мобильных ML-фреймворков, позволяющие выбрать наиболее подходящее решение в зависимости от требований к производительности, энергоэффективности и размеру мобильного приложения.

С учетом быстрого развития технологий и растущего спроса на интеллектуальные мобильные приложения, а также, нынешних ограничения связи, из-за которых невозможно полагаться на серверные вычисления, для стабильной работы приложения, перспективы развития данной области крайне высоки.

Использованные источники

1. Zhang J., Zhang Y., Lu Y. A comparative study of deep learning frameworks based on short-term power load forecasting experiments // Journal of Physics: Conference Series. 2021. Vol. 2005. No. 1. Article 012070. DOI: 10.1088/1742-6596/2005/1/012070..

2. Jocher G. Comparative Analysis of YOLOv11 Deployment Options [Электронный ресурс] // Ultralytics. URL: https://docs.ultralytics.com/ru/guides/model-deployment-options/#comparative-analysis-of-yolo11-deployment-options_1 (дата обращения: 15.05.2026).

3. Mishra A. TensorFlow Lite vs PyTorch Mobile [Электронный ресурс] // Analytics Vidhya. 10.12.2024. URL: https://www.analyticsvidhya.com/blog/2024/12/tensorflow-lite-vs-pytorch-mobile/ (дата обращения: 15.05.2026).

4. Zhang Q. et al. A comprehensive benchmark of deep learning libraries on mobile devices // Proceedings of the ACM Web Conference 2022. New York: ACM, 2022. P. 3298–3307. DOI: 10.1145/3485447.3512125.